Plagiarism Detection Based on Structural Information

Efstathios Stamatatos
Dept. of Information and Communication Systems Eng.
University of the Aegean, Greece
stamatatos@aegean.gr

ABSTRACT
In this paper a novel method for detecting plagiarized passages in document collections is presented. In contrast to previous work in this field that uses mainly content terms to represent documents, the proposed method is based on structural information provided by occurrences of a small list of stopwords (i.e., very frequent words). We show that stopword *n*-grams are able to capture local syntactic similarities between suspicious and original documents. Moreover, an algorithm for detecting the exact boundaries of plagiarized and source passages is proposed. Experimental results on a publicly-available corpus demonstrate that the performance of the proposed approach is competitive when compared with the best reported results. More importantly, it achieves significantly better results when dealing with difficult plagiarism cases where the plagiarized passages are highly modified by replacing most of the words or phrases with synonyms to hide the similarity with the source documents.

Categories and Subject Descriptors
H.3.1 [**Content Analysis and Indexing**]

General Terms
Algorithms, Experimentation.

Keywords
Plagiarism detection, Stopwords, *n*-grams.

1. INTRODUCTION
A side effect of the rapid growth of online publishing of text in Internet media is that plagiarism became easier than ever. Plagiarism is particularly evident in journalism (i.e., newspapers, blogs) and academia (i.e., student reports, theses) [7]. As a result, significant parts or even entire documents are exact copies or modifications of a single or multiple original sources. While many plagiarism cases are easy to be found by human readers, the great volumes of suspicious and source texts demand automatic plagiarism detection tools to facilitate this process.

Automatic plagiarism detection comprises several tasks. The default scenario (aka *external plagiarism detection*) regards the identification of passages in suspicious documents as likely plagiarized and associate them with certain passages of source documents in a given reference collection [18,20]. *Intrinsic plagiarism detection* considers the case where no reference collection is available and the likely plagiarized passages in a suspicious document have to be extracted based on stylistic inconsistencies [23]. *Cross-lingual plagiarism detection* deals with the case where the suspicious and source documents are written in different natural languages [17]. *Text reuse* or *near-duplicate detection* is associated with plagiarism detection since it attempts to find documents that share most of their content and are derivatives of an original source [4,10]. However, it examines similarity on the document level. *Local text reuse* or *partial-duplicate detection* is closer to plagiarism detection where a very short passage may be copied in a long document [22,27]. In this task, the similarity is considered legitimate, so usually there is no attempt to hide it. As a result, it resembles the verbatim case of plagiarism detection.

One major concern in plagiarism detection is efficiency [21]. The suspicious documents should be exhaustively compared with any document in the reference collection which may be very large (i.e., the whole indexed Web). It is therefore necessary for the similarity estimation between a pair of documents to be based on simple measures. Additionally, plagiarism detectors should be able to capture local similarities where only a likely short passage is common in both documents. Given that the plagiarized and the original passages may not be exactly the same in case the plagiarist performed some kind of paraphrasing, the information used to represent texts should capture the similarity even when many words and word ordering are different [9].

Existing approaches in plagiarism detection are based on sequences of words or characters to represent texts [2,10,21]. The content words (e.g., nouns, adjectives, proper-names, etc.) are generally considered the most important information to identify local similarity in texts. Very frequent words conveying no meaning (i.e., stopwords) are usually excluded [6,9,10,18] or used to identify the position of important content terms [25].

It is a common practice in Information Retrieval (IR) to discard stopwords since they increase the size of index with many word occurrences. According to the *rule of 30*, the 30 most common words account for (roughly) 30% of the word tokens in a corpus [14]. However, efficient index compression methods can considerably decrease the size required by these occurrences. Moreover, the elimination of stopwords makes phrase queries more difficult or even impossible to be processed. As a result, modern IR systems, including many Web search engines, adopt full-text indexing [14]. Stopwords have been proved to be extremely useful in text mining tasks including authorship attribution [1] and text-genre detection [24] where the aim is to represent style rather than content. In plagiarism detection, it has been demonstrated that stopword removal considerably hurts the performance [6].

In this paper, we propose a novel plagiarism detection method that is based on structural rather than content information. Instead of following the common practice of eliminating stopwords, the

CIKM'11, October 24–28, 2011, Glasgow, Scotland, UK.

proposed method eliminates all the other tokens and is entirely based on the remaining stopword sequences to represent the syntactic structure of texts. It is shown that stopword *n*-grams are robust and accurate when used to capture local similarities between suspicious and original documents. Moreover, we propose an algorithm for detecting the exact plagiarized passage boundaries in suspicious and source documents. The presented results on a publicly-available corpus demonstrate that the performance of the proposed approach is competitive when compared with the best reported results on the same corpus. More importantly, our method achieves significantly better results when dealing with difficult plagiarism cases where the plagiarized passages are highly modified and most of the words or phrases have been replaced with synonyms.

The rest of this paper is organized as follows. The next section describes previous related work. Section 3 presents the proposed method in detail. The experimental settings and results are included in Section 4 while the conclusions drawn from this study and suggested future work directions are given in Section 5.

2. RELATED WORK

The majority of approaches to plagiarism detection adopt the same architecture [18]. First, to improve efficiency in large document collections, for each suspicious document a small set of candidate source documents is retrieved. This set is either of predefined or variable size according to the similarity between the documents. Then, a more detailed analysis between the suspicious document and each of the retrieved documents provides the requested passage boundaries. Finally, a post-processing step checks these detections and removes or merges some of them.

In order to detect the degree of similarity between documents, two basic approaches have been proposed. The first follows the typical IR methodology that considers the suspicious document (or parts of the document) as a query and attempts to rank documents in the reference collection according to their similarity with the query [9,10,16]. The similarity measures take into account relative word frequencies, document frequencies, and document lengths [15] while stopwords are usually discarded [9,10]. To take into account word substitutions by synonyms Gustafson et al. [9] proposes the use of word-correlation factors that measure frequency of co-occurrence and relative distance between pairs of terms in Wikipedia documents. The syntactic structure of sentences is more robust in cases of paraphrasing the plagiarized passages [26] but the required syntactic analysis considerably harms the efficiency.

The second basic family of approaches relies on document fingerprints comprising hashes of fixed-length chunks (aka *shingles*) in documents [21,22]. The complete set of chunks can be included in the document fingerprint (full fingerprinting) to optimize effectiveness or, alternatively, a chunk selection method can be applied to decrease storage requirements and optimize efficiency [21]. Some approaches define chunks so that to capture information about the content and the structure of a short piece of text. Usually they are character *n*-grams [21], word *n*-grams [2] or sentences [8,27]. Word *n*-grams can be sorted to be more flexible in small changes between the plagiarized and the source passages [11]. Theobald, et al., [25] use stopword positions to identify useful chains of content words in web pages. In contrast, Basile, et al. [3] consider chunks that are based on word-length sequences excluding any content information.

Provided a suspicious document is found to be similar with a source document, a scatter plot of the positions of all the matches found between the two documents can reveal the approximate passage boundaries [27, 28]. In case of verbatim plagiarism or partial-duplicate detection, these passages will be straight diagonal lines. To detect such passage boundaries, algorithms for finding diagonals of maximal length are appropriate [27]. However, in cases when the plagiarized passage is modified there is noise in these diagonal lines. A cluster of matches is produced and it is usual to have small gaps between adjacent areas that correspond to the same passage. To solve this problem, several methods have been proposed including sets of heuristic rules to identify and merge adjacent passages [3,11,12], Monte Carlo optimization to join adjacent matches [8], and application of clustering methods [28]. Although this kind of analysis has to be performed for relatively few source documents per suspicious document, it can harm the efficiency of the approach when its computational cost is high.

After the detection of passage boundaries, the post-processing step is used to filter the passage detections and eliminate or merge cases of short passages and overlapping or ambiguous (e.g., indicating the same plagiarized passage and different source passages) detections [11,12,16,28]. A final verification of similarity between the passages in the suspicious and the source documents has also been proposed [16]. The post-processing step is especially important for improving the precision of the plagiarism detection methods.

Recently, two competitions on plagiarism detection were organized addressing several plagiarism types, including external plagiarism, intrinsic plagiarism, and cross-lingual plagiarism [18, 20]. Evaluation corpora and methodologies have been released [19] providing the possibility to compare different approaches on the same testing ground. The focus of the evaluation in these competitions is on the exact detection of passage boundaries in plagiarized and source documents. Although the majority of the participants eliminated stopwords to increase the efficiency of document representation, the winning methods avoided explicitly removal of stopwords. The winner of the 2009 competition used character 16-grams [8] while the winner of the 2010 competition used (sorted) word 5-grams including all words with at least three characters [11].

3. THE PROPOSED METHOD

In this study, monolingual plagiarism detection is considered. Let D_x be a set of suspicious documents and D_s be the set of source documents (i.e., reference collection). The first task is to decide whether a suspicious document is plagiarized or non-plagiarized. In the former case, all the sources of plagiarism should be identified including source documents (a subset of D_s) and the exact boundaries of the plagiarized passages in both the suspicious and source documents. Furthermore, it is desirable to assign a score to each detected plagiarized passage to indicate the degree of plagiarism. This score can be used to sort the detected passages from exact copies to somehow related passages. The general architecture of the proposed method follows the state-of-the-art in this field [18].

3.1 Text Representation

The representation of texts according to the proposed method is based on stopword *n*-grams (SWNG). Given a document and a list of stopwords, the text is reduced to the appearances of these stopwords in the document. All the other tokens are discarded. As stopwords, in this study, we use a list of the 50 most frequent words of English (Table 1) extracted by the British National Corpus which includes about 90 millions tokens. Therefore, a text is first transformed to lowercase, then it is tokenized and all the tokens not belonging to the list of stopwords are removed. Finally, the *n*-grams of the remaining stopwords are produced. We call this set of SWNGs the *profile* of the document. Given a document *d*, the profile $P(n,d)$ comprises all the stopword *n*-grams, i.e., analogous to the full-fingerprinting method [10]. The SWNGs in $P(n,d)$ are ordered according to their first appearance in the document. The procedure of transforming text passages to a set of stopword *n*-grams is demonstrated in Figure 1.

Table 1. The list of 50 most frequent words of BNC corpus.

1. the	11. with	21. are	31. or	41. her
2. of	12. he	22. not	32. an	42. n't
3. and	13. be	23. his	33. were	43. there
4. a	14. on	24. this	34. we	44. can
5. in	15. I	25. from	35. their	45. all
6. to	16. that	26. but	36. been	46. as
7. is	17. by	27. had	37. has	47. if
8. was	18. at	28. which	38. have	48. who
9. it	19. you	29. she	39. will	49. what
10. for	20. 's	30. they	40. would	50. said

The intuition behind this representation is that stopword occurrences are usually associated with syntactic patterns. Therefore, sequences of stopwords reveal hints of the syntactic structure of the document that is likely to remain stable during the procedure of plagiarizing a passage. That is, when one attempts to plagiarize a particular passage of text and wants to cover their traits, the most usual act is to replace words and phrases with available synonyms. It is much more difficult to change the basic syntactic structure or rewrite large parts of the text. Stopwords are function words, that is they are content-independent and they do not convey any semantic information. They can usually be removed/replaced when the syntactic structure changes.

According to the terminology introduced in the work of Koppel, et al. [13], a language element (i.e., a word or a syntactic structure) is *unstable* when it can be replaced by other semantically equivalent elements. *Stability* of words can be regarded as the availability of synonyms. Given that definition, stopwords are words with high stability and, therefore, are likely to remain intact when someone attempts to slightly modify a text passage. In case the modification does not involve significant reordering of contents, long sequences of stopwords of the original passage are likely to also be included in the modified passage. Moreover, language diversity and language errors especially when the authors are non-native speakers can affect the stability of words. For example, the tokens 'plagiarize',

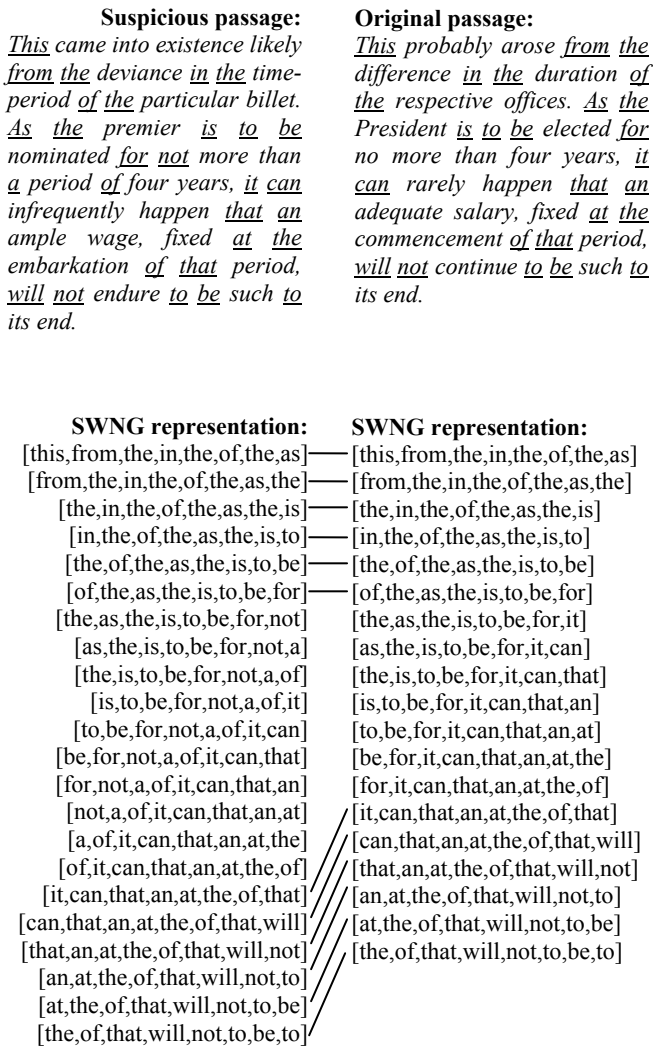

Figure 1. An example of a plagiarism case and the proposed stopword n-gram representation (common 8-grams of the passages are indicated with lines).

'plagiarise', 'pladgiarize', and 'plagarize' are some different (correct or erroneous) versions of the same content word. On the other hand, most speakers of the language are familiar with stopwords and since they are relatively short, they are less likely to contain errors.

The stability of stopwords is demonstrated in the example of Figure 1 where an original piece of text and a plagiarized version of it are given. Despite the fact that the plagiarized version is highly modified, many sequences of our list of 50 stopwords remain the same with those of the original document (the original and the plagiarized passage have 18 common 5-grams, 12 common 8-grams, and 6 common 11-grams of stopwords). This similarity is affected in the case the plagiarist rewrites significant parts of the passage. On the other hand, texts that are not associated are unusual to share long sequences of SWNGs since that would mean they share the same syntactic structure in consecutive sentences or entire paragraphs. The discussion in the following section shows that such coincidental similarity is rare.

3.2 Candidate Retrieval

The first important step in plagiarism detection is the retrieval of a subset of D_s that comprises the sources of likely plagiarism in a suspicious document. This procedure includes the exhaustive comparison of the suspicious document with any member of D_s to identify any local similarities. In general, the number of source documents for a given suspicious document is not known a priori. It could be none, a single, or multiple source documents. The most important issue here is to achieve a high recall since it is just the first step in the detection process and any source document missed will no further examined. On the other hand, a low precision score will affect the efficiency of the next steps.

Given the SWNG representation, our aim is to find common *n*-grams of stopwords between the suspicious and the source documents. The main question here regards the definition of an appropriate value of *n*. How long should the sequences of stopwords be so that to detect a similarity between a suspicious and a source document? Let n_1 be this value. Any common *n*-gram between a pair of documents with $n<n_1$ is considered not significant. A common *n*-gram with $n>=n_1$ suggests a match that is not coincidental. In that sense, the value of n_1 should be relatively high. On the other hand, beyond the case of verbatim plagiarism, when the plagiarized passages have been highly modified, we should not expect to find too long common sequences of stopwords. In those cases, a high value of n_1 would miss source documents including the originals of either short or highly modified plagiarized passages. Therefore, there is a trade-off between low and high values of n_1 for the candidate document retrieval task.

One common case of coincidental similarity between the sequences of stopwords of unrelated documents is when the sequence contains only specific, very frequent stopwords. These words are the first 6 most frequent stopwords (*the*, *of*, *and*, *a*, *in*, *to*) plus *'s*. An example is shown in Figure 2 where two unrelated text passages have exactly the same sequence of stopwords (11-gram). Such cases considerably increase the false positives of our approach. To avoid them, we need an additional constraint on the contents of the common *n*-grams found in the profiles of two documents. This constraint should not be too rigid so that similarities of short plagiarized passages are not filtered out. Let C={*the*, *of*, *and*, *a*, *in*, *to*, *'s*} be the set of the stopwords usually appear in coincidental matches. Let $d_x \in D_x$ and $d_s \in D_s$ while $P(n_1, d_x)$ and $P(n_1, d_s)$ are the corresponding profiles of these documents comprising SWNGs of length n_1. A match between these documents is detected when the following criterion is satisfied:

$$\exists g \in P(n_1,d_x) \cap P(n_1,d_s): \; member(g,C) < n_1 - 1 \; \wedge \; maxseq(g,C) < n_1 - 2 \quad (1)$$

where the functions $member(g,C)$ and $maxseq(g,C)$ return the number of stopwords of the *n*-gram g that belong to C and the maximal (longest) sequence of stopwords of g that belong to C, respectively. For instance, when n_1=11, a match g (i.e., a common 11-gram in the profiles of a suspicious and a source document) would indicate a possible plagiarism case only when g contains at least 2 stopwords not belonging to C (i.e., $member(g,C)<10$) and the longest sequence in g of stopwords belonging to C is less than 9. Note that the example of Figure 2 fails to satisfy both of these constraints since $member(g,C)=10$ and $maxseq(g,C)=10$.

The minutes of the committee record the motion of appreciation to the owners. Mr. Robert Bell of the old printing firm of that name made…

…the Fathers of the Church; the aesthetic mysticism of Plotinus, reborn to its greatest triumphs, during the classic period of German thought. Through the midst of these variously erroneous theories, that traverse…

Figure 2. Two unrelated text passages with the same sequence of stopwords.

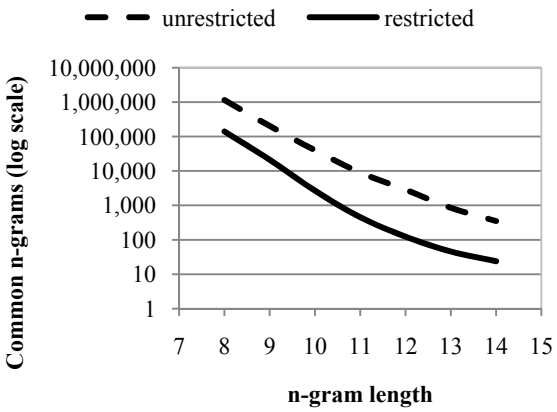

Figure 3. Number of common *n*-grams in a collection of 1,000 documents without any known case of plagiarism before and after applying criterion (1).

Figure 3 depicts the amount of common *n*-grams in a collection of 1,000 documents (a part of the corpus described in Section 4.1) without any known case of plagiarism before and after the application of criterion (1). The document length in this collection varies from 3,000 to 2.5 million characters. Apparently, this criterion significantly reduces the amount of coincidentally common *n*-grams. Note that usually there are many cases where two documents may share the same (short) passage (e.g. famous quotations) [12]. So, some long stopword *n*-grams are likely to be detected in some documents of any collection. This is further discussed in Section 3.4.

3.3 Passage Boundary Detection

When a set of source documents that match a suspicious document has been retrieved, the next step is to perform a more detailed analysis to estimate the exact boundaries of plagiarized passages in both the suspicious and the source documents. Let $D_{rx} \subseteq D_s$ denote the set of source documents that have been retrieved for the suspicious document d_x. Our aim is to find the common SWNGs in the profiles of d_x and each $d_s \in D_{rx}$ and build maximal sequences of them that correspond to text passages.

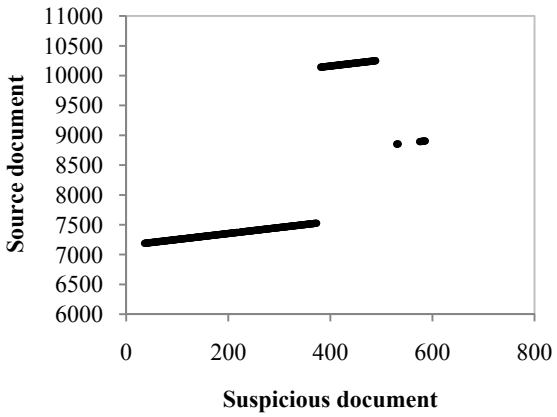

Figure 4. Scatter plot of the matched n-grams in verbatim plagiarism cases where the plagiarized passages are next to each other.

In case the plagiarized passage is an exact copy of the source document, the task is quite easy since exactly the same sequence of SWNGs will be included in both profiles and in the same order. Then, the scatter plot showing the matches between a suspicious and source document will be composed of diagonal lines. An example of verbatim plagiarism cases is given in Figure 4. However, when the plagiarized passage is highly modified there will be considerable noise (i.e., deviation from the diagonal line) and gaps between common SWNGs of the two profiles. The amount of noise and gaps depends on the value of n (order of n-grams) used in producing the profiles of the documents. The higher n is, the more gaps and noise will appear. Therefore, the long n-grams used to identify similarity between documents in the previous step (n_1) are not appropriate in the current step. We need shorter n-grams (of order n_2) so that more detailed matches between the documents to be captured. In order to avoid noise of coincidental matches of SWNGs due to n-grams containing only stopwords of C, we also need a criterion similar to (1) to exclude some uninformative SWNGs. However, to keep the gaps between common SWNGs low, this criterion should be more relaxed in comparison to (1). Let $P(n_2, d_x)$ and $P(n_2, d_s)$ be the profiles of the suspicious and source documents comprising stopword n_2-grams. A n_2-gram g is a match between these documents when the following criterion is satisfied:

$$g \in P(n_2,d_x) \cap P(n_2,d_s) \wedge member(g,C) < n_2 \qquad (2)$$

where the function $member(g,C)$ returns the number of stopwords of the n-gram g that belong to C. Let $M(d_x,d_s)$ be the set of the matched n-grams between the profiles $P(n_2,d_x)$ and $P(n_2,d_s)$ of the suspicious and source documents. Members of $M(d_x,d_s)$ are ordered according to the first appearance of a match in the suspicious document. For example, in the case of the text passages of Figure 2, the ordered set of matches between the 8-grams of the plagiarized and the original passages are indicated with lines. That is, the first 8-gram of the plagiarized passage is identical with the first 8-gram of the original document, the 17th 8-gram of the plagiarized passage is identical with the 14th 8-gram of the original document, etc. Moreover, let M_1 and M_2 be the parts of M that correspond to the suspicious document and the source document, respectively. For instance, in the example of Figure 1, M_1={1, 2, 3, 4, 5, 6, 17, 18, 19, 20, 21, 22} while M_2={1, 2, 3, 4, 5, 6, 14, 15, 16, 17, 18, 19}. Therefore, consecutive M_1 values always increase while consecutive M_2 values may decrease as

```
Input: dx, a suspicious document
       ds, a source document
       n2, length of stopword sequences
       θg, threshold of maximum gap allowed
Output: a set of detections

detectPassageBoundaries(dx,ds,n2,θg)
1. Px=profile(n2,dx);
2. Ps=profile(n2,ds);
3. [M1,M2]=match(Px,Ps);
4. InitPlagPass=findPassages(M1,θg);
5. Detections=[];
6. PlagPassages=[];
7. for all Pi ∈ InitPlagPass
8.    Oi=subset(Pi,M2);
9.    OrigPassages=findPassages(Oi,θg);
10.   if size(OrigPassages)>1
11.      for all Oj ∈ OrigPassages
12.         Pj=subset(Oj,M1)
13.         PlagPassages=PlagPassages ∪ Pj;
14.      endfor
15.   else PlagPassages=PlagPassages ∪ Pi;
16.   endif
17. Detections=Detections ∪
              [PlagPassages, OrigPassages];
18. endfor
19. return Detections;
```

Figure 5. The proposed algorithm for detecting passage boundaries.

well. As shown in Figure 4 (scatter plot of M_1 vs. M_2) the boundaries of plagiarized passages are associated with big changes in consecutive values of M_1 and M_2. However, if these changes are not big enough they may correspond to gaps in noisy cases where the plagiarized passage is heavily modified.

Another important problem in this task is when there are multiple plagiarized passages in a suspicious document and the distance between them is relatively low. This case is depicted in Figure 4, where the plagiarized passages are next to each other in the suspicious document (x-dimension). Note that this is not necessarily related with the distance of the original passages in the source document (y-dimension). Similarly, two original passages in the same source document can be close enough while the distance between the corresponding plagiarized passages in the suspicious document may be high.

To handle this problem in the detection of passage boundaries, we propose the following procedure. First, an initial set of passage boundaries of maximal length is detected in the suspicious document allowing small gaps to be included. Then, the corresponding passages in the source document are examined. In case a passage in the source document is not homogeneous (i.e., it comprises parts of the document with significant gaps between them) it splits into smaller passages. Finally, the passage boundaries in the suspicious document are determined based on these smaller passages of the source document. In more detail, the initial set of passage boundaries in the suspicious document is detected according to the following criterion:

$$m_i \in M_1(d_x,d_s): abs(diff(m_i)) > \theta_g \qquad (3)$$

where the functions *abs* and *diff* return the absolute value and the difference (derivative) and θ_g is a threshold that permits relatively

small gaps to be included in the detected passage. If there are adjacent boundaries, they are joined to a single boundary. Each detected passage in the suspicious document (a subset of M_1 values) corresponds to a subset of M_2 values. However, a subset of M_2 values may correspond to different passages of the original document (i.e., the case depicted in Figure 4). Then, each $M_{2i} \subseteq M_2$, corresponding to a maximal subset of a detected passage in M_1 values, is examined to detect maximal passages of the original document. The boundaries of the source document passages are detected according to the following criterion:

$$m_i \in M_{2i}(d_x,d_s): abs(diff(m_i)) > \theta_g \quad (4)$$

where $M_{2i}(d_x,d_s)$ is a subset of M_2 that corresponds to an already detected plagiarized passage in the suspicious document. Gaps lower than θ_g are allowed in a passage. Again, if there are adjacent boundaries, they are joined to a single boundary. Finally, in case multiple passages are detected in the original document, the corresponding passage in the suspicious document is split accordingly to produce the final boundaries of the plagiarized passages. Note that this procedure detects boundaries in the sequence of *n*-grams. Let $<S_i, E_i>$ be the start and ending *n*-gram boundaries of a detected passage. These can be transformed into character boundaries by taking the position of the first character of the first word of S_i and the position of the last character of the last word of E_i.

The passage boundary detection algorithm is shown in Figure 5. The *profile* function extracts the stopword sequences from a text while *match* finds the matches between two texts according to criterion (2). The *findPassages* function detects passages of maximal length in a sequence of matched *n*-grams following the criteria (3) and (4) to estimate the passage boundaries, *size*(*X*) returns the size of the set *X*, and *subset*(*X*,*Y*) extracts a sequence of matched *n*-grams in document *X* that corresponds to a detected passage in document *Y*. The output detections are composed of the plagiarized passages and their corresponding original passages.

3.4 Post-processing

The procedure described so far, is based on SWNG representation and disregards all the words of the text not belonging to the set of the 50 stopwords. The detections obtained, especially in case they are short, should be checked to verify that the similarity of the detected plagiarized passage with the detected original passage is high, when the full text of the passages is taken into account. Moreover, we need a mechanism to assign scores to the detected plagiarism cases according to the degree of similarity with the original passages. This procedure should not be computationally expensive since it will be applied to full text of multiple passages. Moreover, it should be flexible so that to capture the similarity even in cases where the plagiarized passage is highly modified and contains many different words with respect to the original passage (i.e., the case of Figure 1).

Each detection is a 4-tuple $<t_x, d_x, t_s, d_s>$ that associates a plagiarized passage t_x in a suspicious document d_x with a passage t_s in an original document d_s. The presented approach examines the similarity between these passages by extracting the profile of character *n*-grams of each passage and calculating the amount of common *n*-grams in the two profiles. To normalize the form of the passages, all characters are transformed into lowercase and punctuation marks are removed. Let $P_c(n,t_x)$ and $P_c(n,t_s)$ be the character *n*-gram profiles (where multiple occurrences of the same *n*-gram are replaced by one single occurrence) of the detected passages in the suspicious and the original document, respectively. Then, the similarity between t_x and t_s is calculated as follows:

$$Sim(t_x, t_s) = \frac{|P_c(n_c,t_x) \cap P_c(n_c,t_s)|}{\max(|P_c(n_c,t_x)|,|P_c(n_c,t_s)|)} \quad (5)$$

where $|a|$ is the size of a. Note that in case the $P_c(n_c,t_x)$ and $P_c(n_c,t_s)$ are identical, the similarity measure is 1. This similarity measure resembles the *containment* measure [5]. However, the denominator ensures that if one of the profiles is much longer than the other, the similarity score is considerably reduced. This is especially useful to filter out cases where adjacent passages were erroneously merged. The choice of n_c is associated with the flexibility of the similarity measure. The longer the character *n*-grams are, the more they will be affected by changes in the plagiarized passage with respect to the original passage. Then, in case the similarity score is above a threshold θ_c the detected plagiarism case is considered true. Otherwise, it is removed from the set of detections. For n_c=3, the similarity of the text passages of the highly modified plagiarism case of Figure 1 is 0.59 while the similarity score of the two unrelated passages of Figure 2 is just 0.18.

Another problem that should be faced in the post-processing stage is the existence of many short passages in both the suspicious and source documents that are not plagiarized. Such passages are usually short and refer to famous quotations, sayings, poems, parts of the Bible, etc. [12]. A couple of examples are given below:

…for we have heard Him ourselves, and know that this is indeed the Christ, the Saviour of the world.

…He who of old would rend the oak, Deemed not of the rebound; Chained by the trunk he vainly broke, Alone, how looked he round!"

Ideally, such cases should not be reported as plagiarism acts. However, their identification among the set of detections is very difficult. Since they are usually almost identical in both the suspicious and the source documents, their similarity score would be very high. The same is true for verbatim plagiarism cases. As already mentioned, such passages are usually very short. Therefore, it is possible to apply a threshold θ_L to the length of the detected passages and filter out the vast majority of these. The length threshold is expected to also hurt the recall of the proposed approach since detected plagiarism cases of very short length will also be eliminated. If the aim is to find any similarities between a suspicious document and a set of source documents, no matter if they are plagiarism cases or not, this length threshold should not be applied.

4. EVALUATION

4.1 Corpus

Recently, in the framework of the PAN Workshop series, evaluation campaigns for plagiarism detectors were initiated [18, 20]. A corpus including multiple suspicious and source documents as well as many types of plagiarism cases was released in 2010 [19]. More specifically, the PAN 2010 Plagiarism Competition corpus[1] (PAN-PC-10) comprises 27,073 documents divided into a set of 15,925 suspicious documents and a set of 11,148 source documents. The length of the documents varies from one page to an entire book of several hundred pages. Half (7,972) of the

[1] http://www.uni-weimar.de/cms/medien/webis/research/corpora/pan-pc-10.html

suspicious documents are non-plagiarized. The other half of the suspicious documents contains 68,558 plagiarism cases that were inserted into randomly selected parts of the suspicious documents. Therefore, there are suspicious documents with only one plagiarized passage and other suspicious documents with dozens of plagiarized passages. 70% of the plagiarism cases refer to the external plagiarism detection task and the rest 30% refer to the intrinsic plagiarism detection task (the originals of the plagiarized passages were not taken from the source documents).

The external plagiarism detection cases have been produced either by humans (simulated) or computational tools (artificial) able to obfuscate a passage by replacing words and phrases with synonyms. In the latter case, it is possible to estimate the degree of obfuscation (high, low, or none). Additionally, 14% of the external plagiarism cases were produced by automatic translation tools that used source documents in Spanish and German. Since the proposed approach aims at the monolingual external plagiarism detection task we used the part of the PAN-PC-10 corpus that refers to this, that is, we excluded the suspicious documents with intrinsic or cross-lingual plagiarism cases. Note that each plagiarized document of PAN-PC-10 contains only one type of plagiarism to facilitate the extraction of a sub-corpus with a certain type of plagiarism cases. Some statistics of the corpus we used in this study are shown in Table 2.

4.2 Measures

For evaluating the produced detections, we use the recently proposed measures of *precision*, *recall* and *granularity* on the passage level [19]. In more detail, let S denote the set of plagiarism cases and R denote the set of detections. Then, macro-average precision and recall are defined as follows:

$$Prec(S,R) = \frac{1}{|R|}\sum_{r\in R}\frac{|\cup_{s\in S} s\cap r|}{|r|} \quad (6)$$

$$\mathrm{Rec}(S,R) = \frac{1}{|S|}\sum_{s\in S}\frac{|\cup_{r\in R} s\cap r|}{|s|} \quad (7)$$

where $s\cap r$ is the amount of overlapping characters between s and r when they share at least one character in both the suspicious and the source passage. Otherwise it is 0. These measures give equal weight to each plagiarism case regardless of its length. Additionally, they do not take into account the similarity score assigned by detectors to each plagiarism case.

In plagiarism detection, recall and precision do not give a complete picture of the effectiveness. In case a detector reports overlapping passages for the same plagiarism case or divides a long passage into shorter segments, recall and precision may be affected (increase). Therefore, we need an additional measure that takes these cases into account. Let $S_R \subseteq S$ be the cases detected in R and $R_s \subseteq R$ be the detections regarding the passage s. Then, the granularity measure is defined as follows:

$$\mathrm{Gran}(S,R) = \frac{1}{|S_R|}\sum_{s\in S_R}|R_s| \quad (8)$$

The minimum and ideal granularity value is 1. The larger the granularity is, the more (possibly overlapping) segments are detected for the same plagiarized passage. Precision, recall, and granularity can be combined to a single measure, *plagdet*, defined as follows:

$$\mathrm{plagdet}(S,R) = \frac{F_1}{\log_2(1+\mathrm{Gran}(S,R))} \quad (9)$$

Table 2. Details about the corpus used in this study.

Plagiarism type	Documents	Plagiarism Cases
Simulated	598	2,347
Artificial: High obfuscation	1,337	14,756
Artificial: Low obfuscation	1,354	14,883
Verbatim	1,728	17,423
Non-plagiarized	7,972	0
Total	12,989	49,409

Table 3. The parameter values used in this study.

Parameter	Value	Function
n_1	11	Stopword n-gram length to retrieve candidate documents
n_2	8	Stopword n-gram length to detect passage boundaries
n_c	3	Character n-gram length to measure similarity between passages
θ_g	100	Upper threshold (in SWNGs) of gap-length allowed in a passage
θ_c	0.5	Lower threshold of the similarity measure to keep a detection
θ_L	200	Lower limit (in characters) of the detected passage length

Table 4. Evaluation results for each processing step.

	Prec.	Recall	Gran.	Plagdet
Candidate retrieval	0.38	0.91	-	-
Passage boundary detection	0.23	0.85	1.02	0.36
Post-processing	0.94	0.83	1.01	0.88

where F_1 is the harmonic mean of precision and recall. Note that the *plagdet* measure was used to rank the candidates in the PAN competitions on plagiarism detection [18,20].

4.3 Results

To apply the presented approach to PAN-PC-10 corpus, a small part of it was first used to estimate the appropriate parameter settings. In more detail, the first 100 suspicious documents (containing non-plagiarized and plagiarized documents of any kind) and their corresponding source documents were used and various values for n-gram length and thresholds were tested. Our aim in these preliminary experiments was not to optimize the results for this specific sub-corpus but to estimate general parameter values that increase recall of the first steps and precision of the last steps. The parameter settings shown in Table 3 were selected and used in the experiments described below.

First, we examine the performance in each processing step. Table 4 shows the results after applying the candidate document retrieval, the passage boundary detection and the post-processing tasks. Note that, for the candidate retrieval task, recall and precision are calculated on the document level while granularity and *plagdet* are not defined. The final precision is very high while recall is lower indicating that many plagiarism cases are not

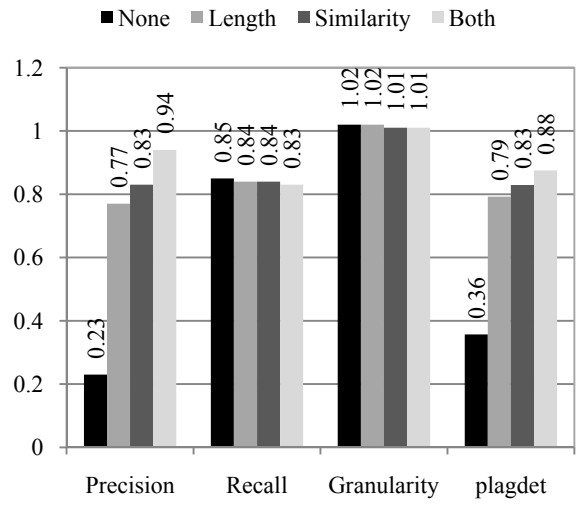

Figure 6. The contribution of the post-processing criteria (length threshold and similarity threshold) to the performance of the presented method.

detected but the provided detections are usually correct. Granularity remains low indicating that in the vast majority of the cases one passage is detected per plagiarism case. The first two steps achieve poor precision scores. However, the post-processing step significantly improves precision.

A more detailed look in the usefulness of the post-processing step is depicted in Figure 6. The performance attained by applying the similarity threshold and the length threshold separately or in combination is given. Apparently, each of these criteria is very important to significantly improve precision. This means that the vast majority of the wrong predictions of the passage boundary detection step correspond to very short passages with similar sequence of stopwords but essentially different content. The combination of these criteria further improves precision due to the elimination of short near-identical passages in suspicious and source documents that are not plagiarism cases (quotations, sayings, etc). Granularity is also improved. On the other hand, recall is slightly reduced.

Next, we examine the performance of the proposed approach in detecting certain plagiarism types. Table 5 shows the results when only simulated plagiarism, artificial plagiarism with high obfuscation, artificial plagiarism with low obfuscation, and verbatim cases are considered. For each type, we use the documents containing this kind of plagiarism (see Table 2) plus an equal number of non-plagiarized documents. This procedure was also followed in [18], so the presented results are directly compared with the performance of the participants in the PAN-10 plagiarism detection competition. The four top-performing participants are denoted as: PAN-10-1 [11], PAN-10-2 [28], PAN-10-3 [16], and PAN-10-4 [8]. The latter was the winner of the PAN 2009 competition using the same method in both competitions. As can be seen, the proposed approach is very competitive in all plagiarism types. It achieves better precision results in any case in comparison to the PAN-10 participants. On the other hand, recall is usually lower in comparison to top-performing approaches. Interestingly, in the most difficult cases of simulated plagiarism and artificial plagiarism with high obfuscation the attained performance is considerably better than the other approaches. This shows that the SWNG representation is better able to capture the structure of a text that remains roughly the same despite significant changes to hide the origin of the plagiarized passages.

5. CONCLUSIONS

Plagiarism detection in large document collections should be both efficient and effective. The former requires that the measures used to represent documents are easily available and capture local similarities so that to enable the identification of a short plagiarized passage within a long document. Moreover, the document representation measures should be flexible in modifications intentionally made by plagiarists to hide the similarity with the original passages. In contrast to the vast majority of the existing approaches that are (entirely or in part) based on content terms to represent documents, in this paper we presented a method that uses structural information acquired by a small list of stopwords.

It has been demonstrated that the stopword *n*-gram method is reliable when it is used to identify similarity in the document level. In addition, an algorithm for detecting exact passage boundaries was proposed. Its performance on a publicly-available corpus is competitive to state-of-the-art approaches achieving higher precision and slightly lower recall results. Interestingly, the proposed method achieves significantly better performance when it deals with difficult plagiarism cases where the plagiarized passage has been extensively modified. In such cases, usually many content words/phrases are replaced by synonyms. This act does not dramatically affect the main syntactic structure of the sentences and, consequently, many stopword sequences remain stable. Note that in these difficult plagiarism cases, content-based methods either cannot capture the similarity or require a more elaborate (and inefficient) analysis of texts. Obviously, when the plagiarist just borrows the ideas of some source documents and rewrites large parts of the passages, the stopword sequences are also affected. In such difficult cases, rare content terms or proper names seem more appropriate to capture the similarity between documents.

The proposed method is very easy to follow and requires minimal resources and text pre-processing cost. The parameter settings used in this paper were obtained manually using a small part of a heterogeneous corpus (i.e., in terms of document genre, document length, plagiarized passage length, and type of plagiarism). In case a more homogeneous corpus is available, machine learning techniques can also be used to acquire a more appropriate set of parameter values for that specific corpus. In this paper, we followed the full-fingerprinting approach where all the stopword *n*-grams are included in the fingerprint of a document. However, techniques that select a subset of stopword *n*-grams can also be applied to reduce the storage requirements and increase efficiency in very large document collections [21].

Provided that modern IR systems adopt full-text indexing, the presented method indicates an additional exploitation of the available information about stopword occurrences. Beyond the improvement in phrase queries, stopword occurrences can also be used to detect likely plagiarism cases. The proposed method can also be applied to detect near-duplicates. The presented document representation based on structural information can be combined with content information to improve the results in difficult plagiarism cases where significant parts of the plagiarized passage have been restructured. An open question regards the minimum

Table 5. Comparative performance results for several plagiarism types.

Plagiarism Type		SWNG	PAN-10-1	PAN-10-2	PAN-10-3	PAN-10-4
Simulated	Prec.	**0.89**	0.33	0.19	0.19	0.33
	Rec.	**0.27**	0.18	0.22	0.26	0.25
	Gran.	**1.00**	**1.00**	**1.00**	**1.00**	1.03
	plagdet	**0.41**	0.23	0.20	0.22	0.28
Artificial: High	Prec.	**0.97**	0.93	0.76	0.77	0.85
	Rec.	0.79	0.75	0.76	**0.81**	0.61
	Gran.	1.03	**1.00**	1.02	1.08	1.02
	plagdet	**0.85**	0.83	0.75	0.75	0.70
Artificial: Low	Prec.	**0.95**	0.93	0.81	0.78	0.82
	Rec.	0.84	**0.92**	0.85	**0.92**	0.66
	Gran.	**1.00**	**1.00**	1.22	1.10	1.01
	plagdet	0.89	**0.92**	0.72	0.79	0.73
Verbatim	Prec.	**0.96**	0.94	0.78	0.76	0.82
	Rec.	0.93	**0.96**	0.86	0.92	0.68
	Gran.	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**
	plagdet	0.94	**0.95**	0.82	0.83	0.74

number of stopwords required to provide accurate results. Moreover, the contribution of each stopword in the detection procedure should be examined so that to form a list of the most effective stopwords for plagiarism detection. This is a language-dependent procedure since it concerns the definition and use of stopwords.

6. REFERENCES

[1] Arun, R., Suresh, V., and Madhavan, C.E.V. 2009. Stopword graphs and authorship attribution in text corpora. In *Proceedings of the IEEE International Conference on Semantic Computing*, 192-196.

[2] Barrón-Cedeño, A., and Rosso, P. 2009. On automatic plagiarism detection based on n-grams comparison. In *Proceedings of the 31th European Conference on IR Research on Advances in Information Retrieval*, pp. 696-700.

[3] Basile, C., Benedetto, D., Caglioti, E., Cristadoro, G., and Esposti, M.D. 2009. A plagiarism detection procedure in three steps: Selection, matches and "squares". In *Proceedings of the 3rd Workshop on Uncovering Plagiarism, Authorship and Social Software Misuse*, pp. 19–23.

[4] Bendersky, M. and Croft, W.B. 2009. Finding text reuse on the web. In *Proceedings of the 2nd International Conference on Web Search and Web Data Mining*, 262-271.

[5] Broder, A.Z. 1997. On the resemblance and containment of documents. In *Proceedings of the Compression and Complexity of Sequences*, 21-29.

[6] Ceska, Z. and Fox, C. 2009. The influence of text pre-processing on plagiarism detection. In *Proceedings of the Int. Conf. on Recent Advances in Natural Language Processing*, 55-59.

[7] Clough, P. 2003. *Old and New Challenges in Automatic Plagiarism Detection*. National UK Plagiarism Advisory Service.

[8] Grozea, C., Gehl, C., and Popescu, M. 2009. ENCOPLOT: Pairwise sequence matching in linear time applied to plagiarism detection. In *Proceedings of the 3rd Workshop on Uncovering Plagiarism, Authorship and Social Software Misuse*, 10-18.

[9] Gustafson, N., Pera, M.S., and Ng, Y.K. 2008. Nowhere to hide: Finding plagiarized documents based on sentence similarity. In *Proceedings of the IEEE/WIC/ACM Int. Conference on Web Intelligence and Intelligent Agent Technology*, 690-696.

[10] Hoad, T.C. and Zobel, J. 2003. Methods for identifying versioned and plagiarized documents. *Journal of the American Society for Information Science and Technology*, 54(3), 203-215.

[11] Kasprzak, J. & Brandejs, M. 2010. Improving the reliability of the plagiarism detection system - Lab report for PAN at CLEF 2010. In *Proceedings of the 4th Workshop on Uncovering Plagiarism, Authorship, and Social Software Misuse*.

[12] Kolak, O. and Schilit, B.N. 2008. Generating links by mining quotations. In *Proceedings of HT 2008*, 117–126.

[13] Koppel, M., Akiva, N., and Dagan, I. 2006. Feature instability as a criterion for selecting potential style markers. *Journal of the American Society for Information Science and Technology*, 57(11), 1519–1525.

[14] Manning, C.D., Raghavan, P., and Schütze, H. 2008. *Introduction to Information Retrieval*. Cambridge University Press.

[15] Metzler, D., Bernstein, Y., Croft, W.B., Moffat, A., and Zobel, J. 2005. Similarity measures for tracking information flow. In *Proceedings of the ACM Conference on Information and Knowledge Management*, 517-524.

[16] Muhr, M., Kern, R., Zechner, M., and Granitzer, M. 2010. External and intrinsic plagiarism detection using a cross-lingual retrieval and segmentation system - Lab report for PAN at CLEF 2010. In *Proceedings of the 4th Workshop on Uncovering Plagiarism, Authorship, and Social Software Misuse*.

[17] Potthast, M., Barrón-Cedeño, A., Stein, B., and Rosso, P 2011. Cross-language plagiarism detection. *Language Resources & Evaluation*, 45(1), 45-62.

[18] Potthast, M., Barrón-Cedeño, A., Eiselt, A., Stein, B., and Rosso, P. 2010. Overview of the 2nd international competition on plagiarism detection. In *Proceedings of the 4th Workshop on Uncovering Plagiarism, Authorship, and Social Software Misuse*.

[19] Potthast, M., Stein, B., Barrón-Cedeño, A., and Rosso, P. 2010. An evaluation framework for plagiarism detection. In *Proceedings of the 23rd International Conference on Computational Linguistics*.

[20] Potthast, M., Stein, B., Eiselt, A., Barrón-Cedeño, A., and Rosso, P. 2009. Overview of the 1st international competition on plagiarism detection. In *Proceedings of the 3rd Workshop on Uncovering Plagiarism, Authorship, and Social Software Misuse*, 1-9.

[21] Schleimer, S., Wilkerson, D.S., and Aiken, A. 2003. Winnowing: Local algorithms for document fingerprinting. In *Proceedings of the ACM SIGMOD International Conference on Management of Data*, 76-85.

[22] Seo, J., and Croft, W.B. 2008. Local text reuse detection. In *Proceedings of the 31st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, 571-578.

[23] Stamatatos, E. 2009. Intrinsic plagiarism detection using character n-gram profiles. In *Proceedings of the 3rd Int. Workshop on Uncovering Plagiarism, Authorship, and Social Software Misuse*.

[24] Stamatatos, E., Fakotakis, N., and Kokkinakis, G. 2000. Text genre detection using common word frequencies. In *Proceedings of the 18th Int. Conf. on Computational Linguistics*, 808-814.

[25] Theobald, M., Siddharth, J., and Paepcke, A. 2008. Spotsigs: Robust and efficient near duplicate detection in large web collections. In *Proceedings of the 31st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, 563–570.

[26] Uzuner, O., Katz, B., and Nahnsen, T. 2005. Using syntactic information to identify plagiarism. In *Proceedings of the ACL Workshop on Educational Applications*, 37–44.

[27] Zhang, Q., Zhang, Y., Yu, H., and Huang, X. 2010. Efficient partial-duplicate detection based on sequence matching. In *Proceedings of the 33rd Int. ACM SIGIR Conference on Research and Development*, 675-682.

[28] Zou, D. Long, W., and Ling, Z. 2010. A cluster-based plagiarism detection method - Lab report for PAN at CLEF 2010. In *Proceedings of the 4th Workshop on Uncovering Plagiarism, Authorship, and Social Software Misuse*.